

# Methods of information systems synthesis

УДК 004.728 : 519.8

doi: 10.20998/2522-9052.2018.3.10

І. С. Зиков<sup>1</sup>, Н. Г. Кучук<sup>2</sup>, С. І. Шматков<sup>2</sup><sup>1</sup> Національний технічний університет “Харківський політехнічний інститут”, Харків, Україна<sup>2</sup> Харківський національний університет імені В. Н. Каразіна, Харків, Україна

## СИНТЕЗ АРХІТЕКТУРИ КОМП'ЮТЕРНОЇ СИСТЕМИ УПРАВЛІННЯ ТРАНЗАКЦІЯМИ E-LEARNING

В роботі розглянуто задачу синтезу архітектури сучасної комп'ютерної системи для управління транзакціями e-learning. **Мета статті** – розробка декомпозиційного методу синтезу архітектури комп'ютерної системи управління транзакціями e-learning на основі стратифікації орієнтованого графа варіантів реалізації комп'ютерної системи. **Результати дослідження.** Проаналізовано основні етапи синтезу архітектури. Обґрунтовано необхідність проведення попередньої декомпозиції, структуризації і формалізації системи. Наведено основні етапи ієрархічної послідовності процесу декомпозиції. Розроблено декомпозиційний метод синтезу архітектури комп'ютерної системи на основі стратифікації орієнтованого графа варіантів реалізації комп'ютерної системи. Розроблено імітаційну модель даного процесу. Результати моделювання показали доцільність застосування розробленого методу при синтезі архітектури комп'ютерної системи з великою кількістю вузлів. **Висновки.** В роботі розглянуто задачу синтезу архітектури сучасної комп'ютерної системи управління транзакціями e-learning. Проаналізовано основні етапи синтезу архітектури. Обґрунтовано необхідність проведення попередньої декомпозиції, структуризації і формалізації системи. Наведено основні етапи ієрархічної послідовності процесу декомпозиції. Розроблено декомпозиційний метод синтезу архітектури комп'ютерної системи на основі стратифікації орієнтованого графа варіантів реалізації комп'ютерної системи. Розроблено імітаційну модель даного процесу. Результати моделювання показали доцільність застосування розробленого методу при синтезі архітектури комп'ютерної системи з великою кількістю вузлів. Так, при  $N < 60$  застосування даного методу недоцільно. При кількості вузлів комп'ютерної системи  $60 < N < 100$  практичного виграшу за часом синтезу не досягається. Однак, при  $N > 100$  виграш стає помітним. При цьому зі збільшенням кількості вузлів перевага декомпозиційного методу наростає.

**Ключові слова:** e-learning; комп'ютерна система; транзакція; синтез; система управління; декомпозиція; стратифікація.

### Вступ

Електронне навчання (e-learning) - це передача знань і управління процесом навчання за допомогою нових інформаційних і телекомунікаційних технологій. Технології e-Learning дозволяють створювати електронні курси, системи управління навчанням та навчальними об'єктами, системи контролю знань, а також засоби створення знань. Електронне навчання може здійснюватися на основі використання цифрових репозитаріїв – баз даних і знань, що дозволяють зберігати знання в електронному вигляді, структуровані відповідно до принципів об'єктного опису метаданими. Цифрові репозитарії дозволяють будувати взаємозв'язки між об'єктами знань з можливістю контекстного пошуку [1, 2].

При побудові або розвитку e-learning необхідно синтезувати архітектуру комп'ютерної системи управління транзакціями. Завдання синтезу архітектури сучасної комп'ютерної системи (КС) для управління складним розподіленим об'єктом (СРО) є комплексною проблемою. Її рішення неможливо без попередньої декомпозиції, структуризації і формалізації. До теперішнього часу багатьма авторами описаний ряд можливих підходів і варіантів розв'язання проблеми [3, 4]. Відповідно до цих роботами можна виділити такі етапи синтезу архітектури КС:

- синтез організаційної структури системи управління (СУ) об'єктом;
- синтез архітектури КС, яка забезпечує вимоги до функціонування системи;

- синтез архітектури базової комп'ютерної мережі (БКС).

Основним завданням синтезу організаційної структури СУ об'єктом є розробка логічної структури СУ об'єкта. Така структура повинна включати основні складові об'єкта і взаємозв'язку між ними. Дана структура визначить вхідні дані для синтезу архітектури КС, за допомогою якої реалізується управління СРО, спрямоване на вирішення завдань певних класів. Така архітектура є логічною сукупністю необхідних компонентів, стратифікована за відповідними рівнями. Крім того, вона повинна враховувати зв'язку між компонентами і рівнями, які забезпечують обмін як службовою інформацією, так і інформацією користувача.

Синтез архітектури базової комп'ютерної мережі включає розробку фізичної структури для отриманої на попередньому етапі архітектури КС.

Крім того, в процесі синтезу, на кожному з етапів, розподіл сукупності вирішуваних завдань, за сукупністю використовуваних компонентів КС повинен прагнути до екстремуму по заданому критерію якості з обов'язковим урахуванням всіх обмежень, які застосовуються.

### 1. Аналіз проблеми та постановка завдання

Підходи до синтезу архітектури комп'ютерної системи СРО розглядаються в багатьох роботах [5-7]. Відповідно до цих роботами синтез складається із таких основних етапів:

1) формалізація структури і компонент, включаючи визначення необхідної кількості компонент і необхідної кількості рівнів, за якими вони повинні бути розподілені;

2) формалізація взаємозв'язків між компонентами, включаючи відтворення необхідних зв'язків в загальній ієрархії рівнів;

3) формалізація задач, що реалізуються такими КС, включаючи їх оптимальний розподіл по компонентах.

Організаційна структура СУ СРО повинна давати однозначну відповідь на питання розподілу процесів по підсистемах різних рівнів, а також розподілу всієї сукупності можливих функцій і методів їх реалізації по необхідним підсистем. При вирішенні задачі синтезу потрібно визначити:

- множини принципів і методів, які повинні реалізовуватися;

- множини взаємопов'язаних функцій, що реалізуються КС;

- множини компонент КС СРО, взаємопов'язаних між собою і стратифікована за рівнями;

- відображення елементів множини взаємопов'язаних функцій, що реалізуються КС СРО, на множини компонентів базової комп'ютерної мережі.

В [8, 9] пропонуються проводити декомпозицію задачі синтезу. В [10] запропоновано ієрархічна послідовність процесів декомпозиції.

Важливим завданням початкового етапу синтезу є визначення та формалізація вимог до КС СРО [11]. Реалізація вимог системою управління можливо багатьма способами, найбільш зручно формалізуються за допомогою математичного апарату теорії графів [12]. Таким чином, стає можливою формалізація взаємозв'язків для синтезованої КС.

При відображенні безлічі взаємопов'язаних функцій і завдань КС СРО, які задані у вигляді відповідних графів, на безліч компонентів КМ можна отримати орієнтований мультиграф, окремі частини якого відповідають можливим варіантам розподілу функцій по компонентах БКС, а дуги відображають взаємозв'язки між ними.

Оптимальним варіантом реалізації КС буде підграф орієнтованого мультиграфом, який одночасно є допустимим варіантом реалізації організаційної структури СУ СРО. Відповідна модель може бути представлена орієнтованим графом

$$G_A = (V_D, R_F, \Theta), \quad (1)$$

де  $V_D = \{v_d, d = \overline{1, d_0}\}$  – множина можливих варіантів організації даних;  $R_F = \{r_f, f = \overline{1, f_0}\}$  – множина можливих варіантів реалізації функцій;  $\Theta = (\theta_{df}), d = \overline{1, d_0}; f = \overline{1, f_0}; \dim \Theta = d_0 \times f_0$  – матриця, що відображає взаємозв'язок даних і функцій. При цьому  $\theta_{df} = 1$ , якщо для формування множини  $v_d$  використовується функція  $r_f$  і  $\theta_{df} = 0$  інакше.

На графі  $G$  можливе завдання варіантів реалізації даних і способів їх формування. Це дозволяє ві-

дображати варіанти перетворення вхідних даних у вихідні дані з метою вибору оптимального варіанту при заданих умовах оптимальності і обмеження існуючих [13, 14]. Однак велика розмірність графа (1) зазвичай не дозволяє знайти оптимальний варіант за прийнятний час [15]. Тому на практиці використовують наближені методи пошуку варіанту із множини допустимих. В [16] показано, що при стратифікації варіантів реалізації архітектури КС час пошуку зменшується, а точність рішення збільшується.

Тому **метою роботи** є розробка декомпозиційного методу синтезу архітектури комп'ютерної системи управління транзакціями e-learning на основі стратифікації орієнтованого графа варіантів реалізації КС.

## 2. Вирішення завдання

В орієнтованому графі (1) можна виділити два підмножини термінальних вершин:  $S$  – вершини-джерела;  $\mathfrak{Z}$  – вершини-стоки. Тоді, якщо  $N$  – кількість нетермінальних вершин графа (1), то

$$V_D = \bigcup_{k=1}^{N_s} s_k \bigcup_{j=1}^{N_t} t_j \bigcup_{i=1}^N x_i, \quad (2)$$

де  $x_i$  – нетермінальна вершина графа (1),  $s_k \in S$ ;  $t_j \in \mathfrak{Z}$ ;  $N_s = \text{card } S$ ;  $N_t = \text{card } \mathfrak{Z}$ ;

Також введемо множину  $E = V_D \bigcup R_F$ .

Введемо на мережі  $G_A$  відношення наступності елементів таким чином. Нехай елемент  $j$  є наступником елементу  $i$  та позначимо  $j \in \sigma(i)$ , якщо

- 1)  $i$  – дуга, що заходить до вершини  $j$ ;
- 2)  $i$  – вершина, із якої виходить дуга  $j$ .

Інакше, якщо  $i$  – вершина мережі, то  $\sigma(i)$  – множина дуг, що виходять із неї, а якщо  $i$  – дуга, то  $\sigma(i)$  – вершина, в яку ця дуга заходить.

Нехай  $i \in \sigma^{-1}(j)$ , якщо  $j \in \sigma(i)$ , тобто  $\sigma^{-1}(j)$  – множина елементів, для яких елемент  $i$  є наступником. Відзначимо, що відношення наступності елементів  $\sigma$  можна ввести на будь-якому орієнтованому графі без контурів. Мережу  $G_A$  із множиною елементів  $E$ , на якій задано відношення  $\sigma$ , позначимо як  $G_A(E, \sigma)$ .

Визначимо як  $(a, b)$ -шлях послідовність елементів  $l_0 = a, l_1, l_2, \dots, l_k = b$ , що  $\forall i = 1, 2, \dots, k$  елемент  $l_i$  є послідовником  $l_{i-1}$ , а шлях, що з'єднує джерело мережі із її стоком – термінальним шляхом.

Будемо вважати, що елементи мережі задані своїми номерами. Правильною нумерацією елементів (по аналогії із правильною нумерацією вершин) назвемо таку, що якщо  $j \in \sigma(i)$ , то  $i < j$ , де  $i, j$  – номера елементів мережі.

Відмітимо, що елементи кінцевого орграфа  $G_A(E, \sigma)$  без контурів завжди можна правильно пронумерувати. Дійсно, можна побудувати біектив-

не відображення  $\psi$  графа  $G_A(E, \sigma)$  на граф  $G'_A(E, R_F)$ , у якого множина вершин відповідає множині вершин графа  $G_A(E, \sigma)$ , а відношення суміжності вершин  $R_F$  – відношенню наступності елементів  $\sigma$  на графі  $G$ . Відомо, що на графах такого вигляду, як  $F_A(E, R_F)$ , для вершин завжди можна задати правильну нумерацію [17], тобто її можна задати і на множині  $E$  елементів графа  $G_A(E, \sigma)$ , що слідує із бієктивності відображення  $\psi$ .

Змішаним  $(s, t)$ -розрізом або змішаним термінальним розрізом назвемо таку множину  $P$  елементів мережі, що якщо їх видалити з неї, то не знайдеться жодного термінального шляху.

Елемент  $\ell$  назвемо надлишковим в розрізі  $P$ , якщо множина елементів  $\{P \setminus \ell\}$  теж є розрізом. Розріз, який не містить надлишкових елементів, – мінімальний, а розріз, що містить надлишкові елементи є надлишковим. У надлишковому розрізі завжди знайдеться елемент, який можна видалити, не порушуючи властивості розрізу. Зауважимо, що з одного надлишкового розрізу можна іноді утворити кілька мінімальних розрізів, виключаючи різні надлишкові вершини. Надалі змішані термінальні розрізи будемо називати просто розрізами.

Алгоритм перерахування розрізів повинен містити спосіб формування вихідного розрізу, метод утворення нових розрізів з побудованих, процедуру виключення надлишкових елементів та порядок перебору. В якості вихідного розрізу візьмемо множину елементів  $P_0 = \sigma(s)$ . Ясно, що надлишкових елементів в ньому бути не може. У побудованому вихідному розрізі елементи розташовані в порядку зростання номерів.

Із кожного побудованого розрізу  $P = \{\ell_1, \ell_2, \dots, \ell_n\}$  можна сформувати  $\ell \leq n$  нових розрізів. Якщо  $t \in \sigma(\ell_i)$ , то

$$P_i = ((P \setminus \ell_i) \cup \sigma(\ell_i)) \setminus I_i, \quad (3)$$

де  $I_i$  – множина надлишкових елементів у  $P_i$ .

У кожному новозбудованому розрізі впорядкуємо елементи за зростанням номерів, а елементи з однаковими номерами замінимо одним. Усі сформовані розрізи заносимо в загальний список розрізів в лексикографічному порядку за зростанням номерів складових елементів. Зауважимо, що при породженні розрізів із  $P$  заміною різних елементів  $\ell_i$  на  $\sigma(\ell_i)$  кілька разів, може утворюватися один і той же мінімальний розріз, який слід занести в список тільки перший раз.

Щоб по можливості зменшити кількість перестановок при лексикографічному упорядкуванні, доцільно елементи із  $P$  замінити на їх  $\sigma$ -образи в порядку, зворотному тому, як вони розташовані в  $P$ , тобто  $\ell_n, \ell_{n-1}, \dots, \ell_1$ . Крім того, слід кожен сформований мінімальний розріз відразу заносити в спи-

сок таким чином, щоб не порушувалася лексикографічна впорядкованість всього списку.

Завдяки прийнятій правильній нумерації елементів мережі, кожен новозбудований розріз розташовується в лексикографічно упорядкованому списку після того розрізу, за яким він є побудованим.

Розглянемо спосіб виключення надлишкових елементів з кожного породженого із розрізу  $P$ :

$$\tilde{P} = (P \setminus \ell_i) \cup \sigma(\ell_i). \quad (4)$$

В розрізі  $\tilde{P}$  деякий елемент  $r_j$  є надлишковим, якщо усі  $(s, r_j)$ -шляхи або усі  $(r_j, t)$ -шляхи проходять ще крізь який-небудь елемент  $r_k \in P$ . Але для прийнятого способу породження розрізів на  $(s, r_j)$ -шляхах подібних елементів  $r_k$  у розрізі  $\tilde{P}$  бути не може. Тому пошук надлишкових елементів слід проводити враховуючи тільки відрізки  $(r_j, t)$ -шляхів.

Впорядкуємо в породженому розрізі  $\tilde{P}$  елементи за зростанням номерів. Тоді для перевірки елементів  $r_j$  на надмірність можна обмежитися відрізками  $(r_j, t)$ -шляхів, на яких номер останнього елемента не більше, ніж номер останнього елемента в розрізі  $\tilde{P}$ .

Позначимо  $\tilde{P} = \{r_1, r_2, \dots, r_m\}$  – розріз, утворений із розрізу  $P = \{\ell_1, \ell_2, \dots, \ell_n\}$  заміною  $\ell_i$  на  $\sigma(\ell_i)$ . Процедура виключення надлишкових елементів буде мати такий вигляд.

Для всіх  $j$  від 1 до  $m$  виконаємо:

положимо  $\Sigma = \sigma(r_j) = \{y_1, \dots, y_p\}$ .

якщо існує  $y \in \Sigma$  таке, що  $y > r_m$ , то  $r_j$  – не надлишковий елемент в  $\tilde{P}$ ;

якщо  $\Sigma \in \tilde{P}$ , то  $r_j$  – надлишковий елемент  $\tilde{P}$ ;

якщо для всіх  $y \in \Sigma$  буде  $y \leq r_m$ , то створимо  $\phi = \Sigma \setminus (\Sigma \cap \tilde{P})$  та положимо  $\Sigma = \sigma(\phi)$ , після чого перейдемо до п. 2.

Розрізи вибираються зі списку побудованих розрізів у лексикографічному порядку за номерами складових їх елементів. При перегляді розрізу з нього породжуються нові розрізи заміною тих елементів  $\ell_i$ , для яких  $\sigma(\ell_i)$  не містить стоку  $t$ .

Весь алгоритм перерахування мінімальних розрізів має трудомісткість  $O(N^2)$ .

Розглянемо спосіб перетворення списку розрізів в мережу. У формалізованому вигляді це завдання можна сформулювати так.

Є список  $\mu$ , що складається з  $M$  рядків  $\xi_1, \dots, \xi_M$ , та володіє такими властивостями:

рядок  $\xi_i = (a_{i1}, \dots, a_{im})$  містить  $m_i$  неповторюваних символів алфавиту  $A = \{a_i\}_{i=1}^N$ ;

допускається перестановка символів в рядку;

не існує  $\xi_i, \xi_j \in \mu$ , таких, що  $\xi_i \leq \xi_j$ .

Знайдеться хоч одна пара символів  $a_i, a_j$ , для яких не існує рядка  $\xi_k \in \mu$ , такого, що  $a_i \in \xi_k, a_j \in \xi_k$ . У цьому випадку символи  $a_i$  та  $a_j$  несумісні.

Треба побудувати мінімальний за кількістю вершин оргграф  $G'_A(Y, R_F)$  із такими властивостями.

кожній вершині  $y \in Y$  відповідає один символ  $a \in A$ ; позначимо таку вершину  $y_a$ ;

для кожного рядка  $\xi_i \in \mu, i = 1, 2, \dots, M$  існує шлях на графі  $G'_A(Y, R_F)$ , що проходить крізь вершини із символами  $a_{i1}, a_{i2}, \dots, a_{im}$ ; назвемо такий шлях відповідним рядку  $\xi_i$  та позначимо як  $\mu_{\xi_i}$ ;

не існує шляху на графі  $G'_A(Y, R_F)$  такого, що  $x_{a_i} \in \mu$  та  $x_{a_j} \in \mu$ , якщо символи  $a_i$  и  $a_j$  несумісні.

Це означає, що потрібно визначити такий порядок символів в рядках списку, щоб об'єднати максимальну кількість співпадаючих символів в рядках і при цьому несумісні символи не опинилися б в одному рядку. Виберемо трійку символів  $a_i, a_j$  и  $a_k$  таку, що символ  $a_i$  належить одночасно двом рядкам  $\ell_p$  и  $\ell_q$ , котрі містять несумісні символи  $a_j \in S_p$  и  $a_k \in S_q$ . При цьому необхідно розмістити символи у рядках  $\ell_p$  та  $\ell_q$  щоб

$$a_i < a_j, a_i < a_k \tag{5}$$

або  $a_i > a_j, a_i > a_k$ . (6)

Співвідношення (5) запишемо як

$$f = a_i(a_j, a_k), \tag{7}$$

а співвідношення (6) як

$$\bar{f} = (a_j, a_k)a_i. \tag{8}$$

Назвемо форму (7) прямою, а форму (8) – оберненою. Форми  $f_1$  и  $f_2$  назвемо еквівалентними, якщо  $f_1 = a_i(a_j, a_k)$ , а  $f_2 = a_i(a_k, a_j)$ .

Форми  $f_1$  и  $f_2$  назвемо альтернативними за відношенню одна до одної, якщо  $f_1 = \bar{f}_2$  або  $f_2 = \bar{f}_1$ . Надалі,  $f_i$  суперечить  $f_j$ , якщо загальна система нерівностей, створена формами  $f_i$  и  $f_j$ , є несумісною. Тоді форма  $f = a_i(a_j, a_k)$  може суперечити тільки формам вигляду ( $\ell \in (\overline{1, N})$ ):

$$\begin{aligned} f_1 &= a_j(a_i, a_\ell), f_2 = a_k(a_i, a_\ell), \\ f_3 &= (a_j, a_\ell)a_i, f_4 = (a_\ell, a_i)a_k, \end{aligned} \tag{9}$$

відповідно:

$$f \vdash f_1; f \vdash f_2; f \vdash f_3; f \vdash f_4. \tag{10}$$

Для  $\bar{f} = (a_j, a_k)a_i$  помітимо, що

$$\bar{f} \vdash \bar{f}_1, \bar{f} \vdash \bar{f}_2, \bar{f} \vdash \bar{f}_3, \bar{f} \vdash \bar{f}_4. \tag{11}$$

Складемо початкову систему форм  $\Sigma = \Sigma_0$ , що містить прямі форми для всіх подібних трійок символів. Пронумеруємо форми цієї системи (нехай їх буде  $Q$ ). Складемо матрицю протиріч форм, використовуючи співвідношення (10) і (11) та правило:

$$v_{ij} = \begin{cases} 1, & \text{якщо } f_i \vdash f_j; \\ 0, & \text{якщо } f_i \text{ не суперечить } f_j; \\ -1, & \text{якщо } \bar{f}_i \vdash f_j. \end{cases} \tag{12}$$

де  $f_i, f_j \in \Sigma$ .

Введемо допоміжні вектори  $D^+, D^-$  и що складаються із таких елементів:

$$\begin{aligned} d_i^+ &= \sum_{j=1, V_{ij}>0}^Q v_{ij}, d_i^- = \sum_{j=1, V_{ij}<0}^Q v_{ij}, \\ \delta_i &= \sum_{j=1}^Q v_{ij} = d_i^+ + d_i^-. \end{aligned} \tag{13}$$

Будемо змінювати систему форм  $\Sigma$ , замінюючи, де потрібно, складові її форми на альтернативні так, щоб в результаті отримати систему  $\Sigma = \Sigma^*$ , яка містить мінімальну кількість форм, що суперечать одна одній. Показником необхідності заміни форми  $f_i$  на  $f_i^*$  служить величина  $\delta_i > 0$ . При заміні форми  $f_i$  на  $\bar{f}_i$  змінюються  $i$ -й рядок та  $i$ -й стовпець матриці  $V$ , а також деякі компоненти векторів  $D^+, D^-$  та  $\Delta$  за таким правилом:

в  $i$ -му рядку матриці  $V$  міняємо  $v_{ij}$  на  $-v_{ij}$ ,  $j = 1, \dots, Q$ , при цьому

$$\begin{aligned} d_i^+(new) &= d_i^-(old), d_i^-(new) = \\ &= d_i^+(old), \delta_i(new) = -\delta_i(old); \end{aligned} \tag{14}$$

в  $i$ -м стовпці матриці  $V$  міняємо  $v_{ki}$  на  $-v_{ki}$ ,  $k = 1, \dots, Q$ , при цьому

$$\begin{aligned} d_k^-(new) &= d_k^-(old) - v_{ki}(old), \\ d_k^+(new) &= d_k^+(old) - v_{ki}(old), \\ \delta_k(new) &= \delta_k(old) - 2v_{ki}(old). \end{aligned}$$

Поступаючи так само для  $f_i$ , у яких  $\delta_i > 0$ , поступово зменшуємо загальну кількість протиріч в системі  $i\Sigma$ , поки не виявиться, що всі  $\delta_i \leq 0$ ,  $i = 1, \dots, Q$ . При цьому отримуємо систему форм

$\Sigma = \Sigma^*$  із мінімальним числом протиріч. Вона поставить частковий (або повний) порядок на множині символів  $A$ . Це дозволить об'єднати максимальну кількість символів в рядках, щоб побудувати граф необхідного вигляду.

### 3. Аналіз результатів

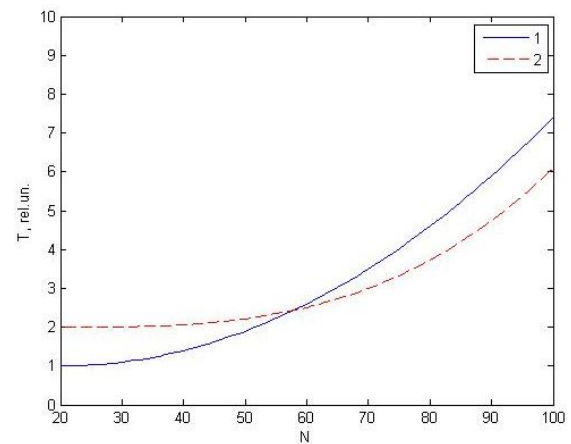
Для аналізу ефективності запропонованого методу синтезу архітектури комп'ютерної системи управління транзакціями e-learning була розроблена імітаційна модель даного процесу. При проведенні моделювання фіксувалися всі вхідні дані моделі, крім кількості вузлів комп'ютерної системи ( $N$ ). Як вихідних даних розглядалося час пошуку оптимального варіанта синтезу комп'ютерної системи. Результати моделювання показані на рис. 1.

Аналіз результатів моделювання показав, що запропонований метод доцільно застосовувати при синтезі архітектури комп'ютерної системи управління транзакціями e-learning з великою кількістю вузлів. Так, графіки рис. 1, а показують незначний програв за часом при використанні запропонованого методу. Очевидна причина цього – часові витрати на стратифікацію графа  $G_A$ . Однак, зі збільшенням кількості вузлів перевага декомпозиційного методу наростає (рис. 1, б, в). Отже, при синтезі архітектури комп'ютерної системи управління транзакціями e-learning на етапі формування зв'язків між компонентами необхідно для зменшення часу процесу застосовувати декомпозиційний метод

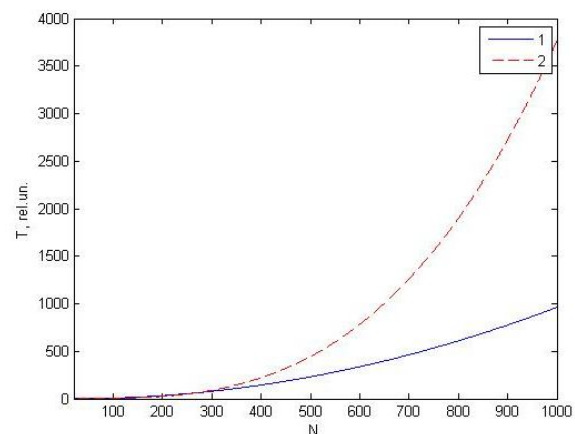
### Висновки

В роботі розглянуто задачу синтезу архітектури сучасної комп'ютерної системи управління транзакціями e-learning. Проаналізовано основні етапи синтезу архітектури. Обґрунтовано необхідність проведення попередньої декомпозиції, структуризації і формалізації системи. Наведено основні етапи ієрархічної послідовності процесу декомпозиції. Розроблено декомпозиційний метод синтезу архітектури комп'ютерної системи на основі стратифікації орієнтованого графа варіантів реалізації КС. Розроблено імітаційну модель даного процесу.

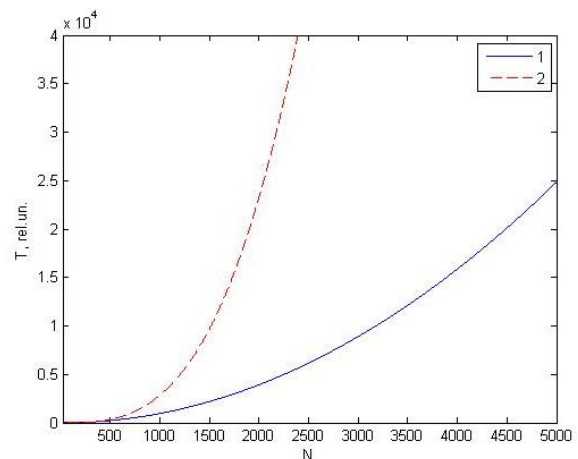
Результати моделювання показали доцільність застосування розробленого методу при синтезі архітектури комп'ютерної системи з великою кількістю вузлів. Так, при  $N < 60$  застосування даного методу недоцільно. При кількості вузлів КС  $60 < N < 100$  практичного виграшу за часом синтезу не досягається. Однак, при  $N > 100$  виграш стає помітним. При цьому зі збільшенням кількості вузлів перевага декомпозиційного методу наростає.



а



б



в

Рис. 1. Залежність часу пошуку оптимального рішення від розмірності мережі: 1 – декомпозиція мережі за рівними стратифікаціями; 2 – без декомпозиції мережі; а – малі системи ( $N < 100$ ); б – середні системи ( $100 < N < 1000$ ); в – великі системи ( $N > 1000$ )

### СПИСОК ЛІТЕРАТУРИ

1. Шматков С. І. Модель інформаційної структури гіперконвергентної системи підтримки електронних обчислювальних ресурсів університетської e-learning / С. І. Шматков, Н. Г. Кучук, В. В. Донець // Системи управління, навігації та зв'язку : науковий журнал. – Полтава : ПНТУ, 2018. – Вип. 2 (48). – С. 97-100.
2. Donets V. Development of software of e-learning information system synthesis modeling process / V. Donets, N. Kuchuk, S. Shmatkov // Сучасні інформаційні системи. – 2018. – Т. 2, № 2. – С. 117-121.
3. Hamed Dinari. A Survey on Graph Queries Processing: Techniques and Methods / Hamed Dinari // Int. Journal of Computer Network and Inf. Security (IJCNIS). – 2017. – Vol. 9, No. 4. – P. 48-56. – URL : <http://dx.doi.org/10.5815/ijcnis.2017.04.06>.

4. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы (4-е издание) / В. Г. Олифер, Н. А. Олифер. – Санкт-Петербург : Питер, 2012. – 943 с.
5. Kuchuk G.A. Method of optimal allocation of relational tables / G.A. Kuchuk, Yu.A. Akimova, L.A. Klimenko // *Engineering Simulation*. – 2000. – Vol. 17. – P. 681-689.
6. Gelenbe E. Analysis and Synthesis of Computer Systems (2nd Edition) / E. Gelenbe, G. Pujolle // *Advances in Computer Science and Engineering*. – 2010. – Vol. 4. – 309 p.
7. Merlac V. Resources Distribution Method of University e-learning on the Hyperconvergent platform / V. Merlac, S. Smatkov, N. Kuchuk, A. Nechausov // *Conf. Proc. of 2018 IEEE 9<sup>th</sup> International Conference on Dependable Systems, Service and Technologies. DESSERT'2018*. Ukraine, Kyiv, May 24-27, 2018. – P. 136-140. – URL : <http://dx.doi.org/10.1109/DESSERT.2018.8409114>.
8. Kuchuk N. Method for calculating of R-learning traffic peakedness / N. Kuchuk; O. Mozhaiev, M. Mozhaiev; H. Kuchuk // *2017 4th International Scientific-Practical Conference Problems of Infocommunications Science and Technology, PIC S and T 2017*. – 2017. – P. 359 – 362. URL : <http://dx.doi.org/10.1109/INFOCOMMST.2017.8246416>.
9. Mukhin V. The Method of Variant Synthesis of Information and Communication Network Structures on the Basis of the Graph and Set-Theoretical Models / V. Mukhin, Yu. Romanenkov, Ju. Bilokin, A. Rohovyi, A. Kharazii, V. Kosenko, N. Kosenko, Ju. Su // *International Journal of Intelligent Systems and Applications (IJISA)*. – 2017. – Vol.9, No.11. – P. 42-51. – URL : <http://dx.doi.org/10.5815/ijisa.2017.11.06>
10. Коваленко А.А. Сучасний стан та тенденції розвитку комп'ютерних систем об'єктів критичного застосування / А.А. Коваленко, Г.А. Кучук // *Системи управління, навігації та зв'язку*. – Полтава . ПНТУ, 2018. – Вип. 1(47). – С. 110-113
11. Mozhaev O. Multiservice network security metric / O. Mozhaev, H. Kuchuk, N. Kuchuk, M. Mykhailo, M. Lohvynenko // *2nd International Conference on Advanced Information and Communication Technologies, AICT 2017 – Proceedings*. – 2017. – P. 133-136.
12. Косенко В. В. Моделирование технической информации та телекомунікаційної мережі на основі її особливого впровадження інформації / В. В. Косенко, Н. Г. Кучук // *Системи обробки інформації*. – Харків: ХУПС, 2016. – Вип. 9 (146). – С. 167-171.
13. Ayodeji J. Limitations of Passively Mapping Logical Network Topologies / Ayodeji J. Akande, Colin Fidge, Ernest Foo // *International Journal of Computer Network and Information Security (IJCNIS)*. – 2017. – Vol. 9, No. 2. – P. 1-11. – URL : <http://dx.doi.org/10.5815/ijcnis.2017.02.01>.
14. Zhang M. Hu. TreePi: A Novel Graph Indexing Method / Zhang, M. Hu, and J. Yang // *IEEE 23rd International Conference on Data Engineering*. – Istanbul, 2007. – P. 966-975.
15. Kuchuk G. Approaches to Selection of Combinatorial Algorithm for Optimization in Network Traffic Control of Safety-Critical Systems / G. Kuchuk, V. Kharchenko, A. Kovalenko, E. Ruchkov // *Proceeding of IEEE East-West Design & Test Symposium (EWDTS'2016)*. – 2016. – P. 384-389. – URL : <http://dx.doi.org/10.1109/EWDTS.2016.7807655>.
16. Кучук Г. А. Управление ресурсами инфокоммуникаций / Г. А. Кучук, Р. П. Гахов, А. А. Пашнев. – Москва: Физматлит, 2006. – 220 с.
17. Xifeng Y. Graph indexing: a frequent structure-based approach / Yan, Xifeng, S.Philip Yu, J.Han // *ACM SIGMOD international conference on Management of data*. – 2004. – P. 335-346.

## REFERENCES

1. Shmatkov, S.I., Kuchuk, N.G. and Donets, V.V. (2018), “Model of information structure of the hyperconvergent system of support of electronic computing resources of university e-learning”, *Control systems, navigation and communication*, Poltava, No. 2 (48), pp. 97-100.
2. Donets, V., Kuchuk, N. and Shmatkov, S. (2018), “Development of software of e-learning information system synthesis modeling process”, *Advanced Information Systems*, Vol. 2, No. 2, pp. 117-121.
3. Hamed, Dinari (2017), “A Survey on Graph Queries Processing: Techniques and Methods”, *International Journal of Computer Network and Information Security (IJCNIS)*, Vol. 9, No. 4, pp. 48-56, available at: <http://dx.doi.org/10.5815/ijcnis.2017.04.06>.
4. Oлифер, V.G., Oлифер, N.A. (2012), *Computer Networks. Principles, Technologies, Protocols*, (4th Edition), St.Petersburg : Piter, 943 p.
5. Kuchuk, G.A., Akimova, Yu.A. and Klimenko, L.A. (2000), “Method of optimal allocation of relational tables”, *Engineering Simulation*, Vol. 17, pp. 681-689.
6. Gelenbe, E. and Pujolle, G. (2010), “Analysis and Synthesis of Computer Systems”, *Advances in Computer Science and Engineering*, Texts. Vol. 4, 309 p.
7. Merlac, V., Smatkov, S., Kuchuk, N. and Nechausov A. (2018), “Resources Distribution Method of University e-learning on the Hyperconvergent platform”, *Conference Proceedings of 2018 IEEE 9<sup>th</sup> International Conference on Dependable Systems, Service and Technologies. DESSERT'2018*, Ukraine, Kyiv, May 24-27, pp. 136-140, available at: <http://dx.doi.org/10.1109/DESSERT.2018.8409114>.
8. Nina Kuchuk; Oleksandr Mozhaiev; Mykhailo Mozhaiev and Heorgii Kuchuk (2017), “Method for calculating of R-learning traffic peakedness”, *2017 4th International Scientific-Practical Conference Problems of Infocommunications Science and Technology, PIC S and T 2017 – Proceedings*, pp. 359 – 362, available at: <http://dx.doi.org/10.1109/INFOCOMMST.2017.8246416>.
9. Mukhin, V., Romanenkov, Yu., Bilokin, Ju., Rohovyi, A., Kharazii, A., Kosenko, V., Kosenko, N. and Su Ju. (2017), “The Method of Variant Synthesis of Information and Communication Network Structures on the Basis of the Graph and Set-Theoretical Models”, *International Journal of Intelligent Systems and Applications (IJISA)*, Vol. 9, No. 11, pp.42-51. available at:<http://dx.doi.org/10.5815/ijisa.2017.11.06>
10. Kovalenko, A.A., Kuchuk, G.A. (2018) “Current state and trends of development of computer systems of objects of critical application”, *Control, navigation and communication systems*. – PNTU, Poltava, No. 1 (47), pp. 110-113.

11. Mozhaev, O., Kuchuk, H., Kuchuk, N., Mykhailo, M. and Lohvynenko, M. (2017), "Multiservice network security metric", *2nd International Conference on Advanced Information and Communication Technologies*, – Proceedings, pp. 133-136.
12. Kosenko, V.V. and Kuchuk, N.G. (2016), "Modeling of Technical Information and Telecommunication Network Based on its Particular Implementation of Information", *Information Processing Systems*, Ivan Kozhedub Kharkiv National Air Force University, Kharkiv, No. 9 (146), pp. 167-171.
13. Ayodeji, J. Akande, Colin, Fidge and Ernest, Foo (2017), "Limitations of Passively Mapping Logical Network Topologies", *International Journal of Computer Network and Information Security (IJCNIS)*, Vol. 9, No. 2, pp.1-11, 2017, available at: <http://dx.doi.org/10.5815/ijcnis.2017.02.01>.
14. Zhang, M. Hu, and Yang J. (2007), "TreePi: A Novel Graph Indexing Method", *IEEE 23rd International Conference on Data Engineering*, Istanbul, pp. 966-975.
15. Kuchuk, G., Kharchenko, V., Kovalenko, A., Ruchkov, E. (2016), "Approaches to Selection of Combinatorial Algorithm for Optimization in Network Traffic Control of Safety-Critical Systems", *Proceeding of IEEE East-West Design & Test Symposium (EWDTS'2016)*, pp. 384-389, 2016, available at: <http://dx.doi.org/10.1109/EWDTS.2016.7807655>.
16. Kuchuk, G.A., Gakhov, R.P. and Pashnev A.A. (2006), *Infocommunication Resources Management*, Fizmatlit, Moscow, 220 p.
17. Xifeng, Yan, Philip S.Yu. and Han J. (2004), "Graph indexing: a frequent structure-based approach", *ACM SIGMOD international conference on Management of data*, pp. 335-346.

Received (Надійшла) 25.06.2018

Accepted for publication (Прийнята до друку) 29.08.2018

### Синтез архитектуры компьютерной системы управления транзакциями e-learning

И. С. Зыков, Н. Г. Кучук, С. И. Шматков

В работе рассмотрена задача синтеза архитектуры современной компьютерной системы для управления транзакциями e-learning. **Цель статьи** – разработка декомпозиционного метода синтеза архитектуры компьютерной системы управления транзакциями e-learning на основе стратификации ориентированного графа вариантов реализации компьютерной системы. **Результаты исследования.** Проанализированы основные этапы синтеза архитектуры. Обоснована необходимость проведения предварительной декомпозиции, структуризации и формализации системы. Приведены основные этапы иерархической последовательности процесса декомпозиции. Разработан декомпозиционный метод синтеза архитектуры компьютерной системы на основе стратификации ориентированного графа вариантов реализации компьютерной системы. Разработана имитационная модель данного процесса. Результаты моделирования показали целесообразность применения разработанного метода при синтезе архитектуры компьютерной системы с большим количеством узлов. **Выводы.** В работе рассмотрена задача синтеза архитектуры современной компьютерной системы для управления сложным распределенным объектом. Проанализированы основные этапы синтеза архитектуры. Обоснована необходимость проведения предварительной декомпозиции, структуризации и формализации системы. Приведены основные этапы иерархической последовательности процесса декомпозиции. Разработан декомпозиционный метод синтеза архитектуры компьютерной системы на основе стратификации ориентированного графа вариантов реализации КС. Разработана имитационная модель данного процесса. Результаты моделирования показали целесообразность применения разработанного метода при синтезе архитектуры компьютерной системы с большим количеством узлов. Так, при  $N < 60$  применение данного метода нецелесообразно. При количестве узлов КС  $60 < N < 100$  практического выигрыша по времени синтеза не достигается. Однако, при  $N > 100$  выигрыш становится заметным. При этом с увеличением количества узлов преимущество декомпозиционного метода нарастает.

**Ключевые слова:** e-learning; компьютерная система; транзакция; синтез; система управления; декомпозиция; стратификация.

### Architecture synthesis of the computer system of transaction control e-learning

I. Zykov, N. Kuchuk, S. Shmatkov

The paper deals with the problem of synthesizing the architecture of a modern computer system for managing e-learning transactions. **The purpose of the article** is the development of a decomposition method for the synthesis of the architecture of the computer system of transaction management e-learning based on the stratification of the oriented graph of options for implementation of the computer system. **Research results.** The main stages of the synthesis of architecture are analyzed. The main stages of the synthesis of architecture are analyzed. The necessity of the preliminary decomposition, structuring and formalization of the system is substantiated. The main stages of the hierarchical sequence of the decomposition process are given. Developed by the deck is a positional method for synthesizing a computer system architecture based on the stratification of an oriented graph of options for implementing a computer system. A simulation model of this process has been developed. The results of the simulation showed the feasibility of applying the developed method in the synthesis of the architecture of a computer system with a large number of nodes. **Conclusions/** The paper considers the problem of synthesis of the architecture of a modern computer system for managing a complex distributed object. The main stages of the synthesis of architecture are analyzed. The necessity of preliminary decomposition, structuring and formalization of the system is substantiated. The main stages of the hierarchical sequence of the decomposition process are presented. A decomposition method is synthesized for the architecture of a computer system based on the stratification of a focused graph of the implementation options of the COP. The simulation model of this process is developed. The simulation results showed the expediency of applying the developed method to the synthesis of the architecture of a computer system with a large number of nodes. Thus, at  $N < 60$  the application of this method is not practical. With the number of nodes COP  $60 < N < 100$ , no practical gain in time for synthesis is not achieved. However, at  $N > 100$  the winning becomes noticeable. Moreover, with the increase in the number of nodes, the advantage of the decomposition method increases.

**Keywords:** e-learning; computer system; transaction; synthesis; control system; decomposition; stratification.